Divide and Conquer: A New Approach to Dynamic Discrete Choice with Serial Correlation

Gregor Reich*

Department of Business Administration
University of Zurich, Switzerland
gregor.reich@uzh.ch

December 24, 2013

Abstract

In this paper, we develop a method to efficiently estimate dynamic discrete choice models with $AR(n)$ type serial correlation of the errors. First, to approximate the expected value function of the underlying dynamic problem, we use Gaussian quadrature, interpolation over an adaptively refined grid, and solve a potentially large non-linear system of equations. Second, to evaluate the likelihood function, we decompose the integral over the unobserved state variables in the likelihood function into a series of lower dimensional integrals, and successively approximate them using Gaussian quadrature rules. Finally, we solve the maximum likelihood problem using a nested fixed point algorithm. We then apply this method to obtain point estimates of the parameters of the bus engine replacement model of Rust [*Econometrica*, 55 (5): 999–1033, (1987)]: First, we verify the algorithm's ability to recover the parameters of an artificial data set, and second, we estimate the model using the original data, finding significant serial correlation for some subsamples.

Keywords: Dynamic discrete choice models; Numerical dynamic programming; Gaussian quadrature; Adaptive grids; Backward iteration.
JEL Classification: C25, C63.

*I am heavily indebted to my advisor Karl Schmedders, and to Ken Judd, for their support and guidance in this project. I would also like to thank Philipp Eisenhauer, János Mayer, John Rust, Ole Wilms, and seminar audiences at the University of Chicago, Hoover Institution, and the University of Zurich for helpful comments and suggestions.

1 Introduction

This paper develops a new approach to efficiently estimate dynamic discrete choice models with $AR(n)$ type serial correlation in the errors: First, we show how to combine some well known methods from the literature, such as Gaussian quadrature, adaptive grid refinement, and methods for large sparse non-linear systems of equations, in order to efficiently approximate the solution to the dynamic optimization problem of the agent. Second, we develop a method to decompose and approximate the integral over the unobserved state variables that appears in the likelihood function, which has previously been considered infeasible for approximation by highly efficient deterministic integration schemes, such as Gaussian quadrature. Finally, we apply the method to the well known bus engine replacement model of Rust (1987) to estimate its parameters in the presence of serially correlated errors by maximum likelihood, using a nested fixed point algorithm. We first apply the method to an artificial data set, in order to verify the algorithm's ability to recover the parameters of the model. Then, we estimate the model using the original data set, and we find significant serial correlation for some of the subsamples of the original dataset.

Dynamic discrete choice models (DDCM) have become a popular instrument for econometric analysis of decision making: First, many (individual) economic decisions we actually can observe are in fact discrete in nature, for example brand or treatment choice. Second, the underlying utility maximization problem of the agents is often dynamic in nature: Not only do the decisions influence their payoffs today, but rather are future decisions and payoffs a result of what choice they make today. By capturing these key facts, DDCM have a wide range of uses; for recent surveys see, for example, Aguirregabiria and Mira (2010) and Keane et al. (2011).

The majority of contributions to the literature on estimation of DDCM makes strong distributional assumptions about the errors, or, used synonymously, the unobserved state variables.[1] Probably the most prominent example is extreme value type I (EV1) iid distributed errors; obviously implied by the EV1 iid assumption, but usually stated explicitly by a conditional independence assumption (CI), the errors are assumed to be serially uncorrelated.

However, there exists a wide consensus that these assumptions are not made because of much empirical evidence, but rather for numerical tractability: EV1 iid errors and CI often induce closed form solutions to potentially high dimensional integrals, arising in the solution to the dynamic optimization problem, the choice probabilities, and the likelihood function. These closed form solutions go back to the work of McFadden (1974, 1981) and Rust (1987).

Several methods exist to estimate DDCM with different distributions of the errors, such as the normal distribution (see Arcidiacono and Ellickson, 2011, in addition to the surveys cited above), or to test the statistical significance of allowing for more general distributions compared to EV1 (Larsen et al., 2012). On the other hand, fewer papers have developed integrated methods to estimate models without the CI assumption, thus allowing for a general notion of serially correlated unobserved state variables. Among them are the simulation and interpolation method of Keane and Wolpin (1994), the Markov chain Monte Carlo approaches of Norets (2009, 2012), and the application of Gaussian quadrature and interpolation as discussed in Stinebrickner (2000).[2]

While the approaches to DDCM estimation with serial correlation are diverse, they share a common challenge:

[1] Throughout the paper, we restrict our attention to continuous unobserved state variables. For discrete unobserved heterogeneity with serial correlation, see, for example, Arcidiacono and Miller (2011).

[2] Other methods estimate DDCM with continuous, serially correlated state variables, but still assume that at least some component of the error is distributed EV1 iid: Hendel and Nevo (2006) approximate the solution to the dynamic problem by policy iteration; similarly, Melnikov (2012) applies value function iteration, but the parameters are estimated using a moment condition rather than maximum likelihood. Finally, Sullivan (2006) develops a method based on Keane and Wolpin (1994) that approximates the solution to the dynamic problem by regressing it on the solution under the EV1 iid assumption.

> "the likelihood function for a DDCM can be thought of as an integral over latent variables (the unobserved state variables). If the unobservables are serially correlated, computing this integral is very hard." (Norets, 2009)

This conclusion follows from the fact that the integral over serially correlated errors really has dimensionality proportional to the time horizon of the data, which itself can be arbitrarily large; moreover, no closed form solution for this integral exists in general.

A popular numerical approach to high dimensional integration is Monte Carlo integration (MC), because its approximation error is independent of the dimensionality of the integral. However, the approximation error usually decreases only very slowly as the number of integration nodes is increased: In order to reduce the estimated error by one order of magnitude, one usually has to increase the number of nodes by two orders of magnitude.[3] Consequently, MC is a natural choice for high dimensional integrals, but only if the integral has no structure that could potentially be exploited by more efficient methods. In contrast to MC, many quadrature rules exist, that have much faster decaying errors, but usually inefficient (in the worst case exponential) scaling in the dimensionality; a popular example is that of Gaussian quadrature rules, extended for multiple dimensions by the product rule.

The approach followed in this paper is to identify and exploit structure that is present in the integral over the unobserved state variables in the likelihood function: Given the serial correlation of the errors is of $AR(n)$ type, the time structure allows us to decompose the high dimensional integral over the time horizon, and rewrite it as a sequence of low dimensional integrals. Then, we can approximate this sequence to high accuracy, using some highly efficient approximation schemes for low dimensional integrals, such as Gaussian quadrature.

In order to evaluate the likelihood function, we need to compute the solution to the dynamic optimization problem of the agent, namely the expected value as a function of the state variables. In the presence of serial correlation, approximating the solution to the dynamic problem involves different numerical tasks: First, taking the expectation of the value function is an integration over the unobserved state variables, for which, in contrast to the EV1 iid case, no closed form solution exists. Consequently, we have to approximate these integrals numerically, and we discuss how to apply Gaussian quadrature, which was first proposed and successfully implemented in the context of DDCM by Stinebrickner (2000). Second, we have to approximate the expected value function as a continuous function of the unobserved state variables; in the EV1 iid case, this step was not necessary, because the unobserved state variables are integrated out in the closed form solution. Different approaches to value function approximation have been proposed (see, for example, Cai and Judd, 2013; Judd, 1998; Rust, 1996), and to stay flexible and generic, we use interpolation over an adaptively refined grid, as proposed by Grüne and Semmler (2004). Third, since the expected value function is only defined implicitly by the fixed point of the dynamic programming operator, we need to solve a non-linear system of equations in order to obtain an approximation of the expected value. While also under the EV1 iid assumption, the expected value is the solution to a fixed point problem, the system becomes much larger in the presence of serial correlation, and thus we discuss suitable methods. Finally, we solve the maximum likelihood problem using a nested fixed point (NFXP) algorithm, which is interconnected with the grid refinement process of the expected value function approximation.

As an application, we estimate the bus engine replacement model of Rust (1987) with serially correlated errors. One motivation for serial correlation in this model is a test for misspecification from the original paper, that leads to the following conclusion:[4]

[3] Formally, the estimated error of the Monte Carlo estimate of some integral I is proportional to $n^{-\frac{1}{2}}$, where n is the number of integration nodes.

[4] One can also think of serial correlation as a feature in this context: In the context of optimal stopping problems, such as the bus engine replacement model, the replacement decision is expected to happen rarely. If the explanatory power of the model in terms of observed states is low, the probability of stopping is small for all possible observed states. Thus, the observed decisions are mostly driven by tail events of the unobserved state

> "for groups 1, 2, and 3 and the combined groups 1-4 there is strong evidence that (CI) does not hold. The reason for rejection in the latter cases may be due to the presence of 'fixed-effects' heterogeneity which induces serial correlation in the error terms." (Rust, 1987)

Testing for statistical significance of serially correlated errors, we find that in some subsamples of the original dataset, we can reject serially uncorrelated errors. Also, the parameter estimates vary substantially, their relative sizes however are rather stable. For readability, the development of the algorithm is closely related to the model under consideration; however, note that it is generic with respect to DDCM with $AR(n)$ type serially correlated errors, that have previously been estimated using the EV1 iid assumption.

The remainder of this paper is organized as follows: Section 2 describes the bus engine replacement model of Rust (1987), and introduces the notion of serial correlation of the errors used throughout the paper. Section 3 first develops a numerical procedure to solve the dynamic programming problem of the agent, then introduces a method to decompose the likelihood function such that it can be computed using highly efficient quadrature rules, and finally describes the likelihood maximization algorithm. Section 4 presents the estimation results. Section 5 concludes.

2 The Bus Engine Replacement Model

In the bus engine replacement model of Rust (1987), an agent repeatedly makes decisions about the maintenance of a fleet of buses: Each period, he observes the state of each of the buses, such as mileage, damages, sign of wear, etc. Based on that, he decides whether to do regular maintenance work only, or a general overhaul; the latter is usually referred to as a replacement of the engine. While the engine replacement causes a fixed cost of RC plus some random component, the cost of regular maintenance is a function $c(\cdot)$ that is increasing in the current mileage state, plus some random component.

Formally, the agent faces single period costs (or negative utility) for each individual bus

$$u_\theta(i, x_t) + \varepsilon_t(i)\,, \quad u_\theta(i, x_t) = \begin{cases} -RC & \text{if } i = 1 \\ -c(x_t, \theta_1) & \text{if } i = 0 \end{cases} \tag{1}$$

where i is the decision variable, with $i = 1$ indicating engine replacement, and $i = 0$ regular maintenance; $\varepsilon_t(i)$ is a random utility component, that is observed by the agent for all possible choices, before making the actual decision. x_t is the mileage of the individual bus at time t, which is reset to 0 after an engine replacement. The replacement cost RC, as well as the cost function parameter θ_1 are both parameters to be estimated. The maintenance cost function is assumed to be of the form $c(x_t, \theta_1) = 0.001\,\theta_1\,x_t$. From the econometrician's point of view, mileage at the time of decision, as well as the decision itself are observable for each bus and each time period. The random utility component however is only observable to the agent, but not to the econometrician; consequently, it is often referred to as unobserved state variable.

For the agent, the decision problem is how long to run a bus with regular maintenance only, with increasing cost induced by increasing mileage, and when to replace its engine, and thus facing the one-time replacement cost, but at the same time reducing the maintenance cost in the future, because mileage is reset to 0. Assuming that the agent behaves dynamically optimal,

variables. However, this fact contradicts the assumption that decisions are modeled to be dynamic, because in a model without serial correlation, these events are unforeseeable, single period shocks. With the introduction of serial correlation, these shocks have persistent effects, which can be anticipated by the agent. For example, a jump in maintenance cost still comes as a surprise to the agent, but once incurred, its effect on future periods can influence decisions to a large extent.

the Bellman equation defines the value per bus as a function of its mileage state and the random utility components

$$V_\theta(x_t, \varepsilon_t) = \max_{i \in \{0,1\}} \{u_\theta(i, x_t) + \varepsilon_t(i) + \beta \mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t]\}. \tag{2}$$

The conditional expectated continuation value in (2) is defined by

$$\mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t] = \int_{(x_{t+1}, \varepsilon_{t+1})} V_\theta(x_{t+1}, \varepsilon_{t+1}) \, Pr(x_{t+1}, \varepsilon_{t+1} \,|\, i, x_t, \varepsilon_t, \theta) \, \mathrm{d}(x_{t+1}, \varepsilon_{t+1}) \tag{3}$$

with subscript θ denoting the dependence of the value function on the parameter values RC and θ_1.

The original model makes the following conditional independence assumption about the joint probability of the state variables:

$$Pr(x_{t+1}, \varepsilon_{t+1} | i, x_t, \varepsilon_t, \theta) = Pr(\varepsilon_{t+1} \,|\, x_{t+1}) Pr(x_{t+1} \,|\, i, x_t) \tag{4}$$

Assumption (4) ensures that (i) the mileage state transition is – conditional on the decision i – independent of the random utility component, and (ii) that the random utility components are serially uncorrelated. If this assumption holds, and if moreover the random utility components $\varepsilon(i)$ are distributed extreme value type I (EV1) iid, the integral in (3) has a closed form solution. However, in order to allow for serial correlation in ε, while keeping (i), we assume

$$Pr(x_{t+1}, \varepsilon_{t+1} \,|\, i, x_t, \varepsilon_t, \theta) = Pr(\varepsilon_{t+1} \,|\, \varepsilon_t, x_{t+1}, \theta) Pr(x_{t+1} \,|\, i, x_t). \tag{5}$$

Note that assumption (5) allows the transition process of the mileage state, $Pr(x_{t+1} \,|\, i, x_t)$, to be estimated independently from the other model parameters – as in the original model.[5] We use discretized mileage, and thus the integral over future mileage states in (3) becomes a sum:

$$\mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t] = \sum_{x_{t+1}} \int_{\varepsilon_{t+1}} V_\theta(x_{t+1}, \varepsilon_{t+1}) \, Pr(\mathrm{d}\varepsilon_{t+1} \,|\, \varepsilon_t, x_{t+1}, \theta) \, Pr(x_{t+1} \,|\, i, x_t) \tag{6}$$

A choice for serial correlation in the unobserved state variables that is frequently used in the literature is the $AR(1)$ process. More specifically, similar to Norets (2009), we define

$$\begin{aligned} \varepsilon_t(0) &= \rho \varepsilon_{t-1}(0) + \tilde{\varepsilon}_t(0)\,, \quad \tilde{\varepsilon}_t(0) \sim q(\cdot) \text{ iid} \\ \varepsilon_t(1) &= \qquad\qquad \tilde{\varepsilon}_t(1)\,, \quad \tilde{\varepsilon}_t(1) \sim q(\cdot) \text{ iid} \end{aligned} \tag{7}$$

where $q(\cdot)$ is a density function with zero mean, and ρ is the additional parameter of the estimation.[6] Thus, for the sake of the argument, we only assume the random utility component of regular maintenance to be serially correlated.[7] It is important to note that definition (7) nests the original model for $\rho = 0$, and $q(\cdot)$ being the density of EV1.

Given that mileage state x_t and decision i_t are observable for all buses, but random utility components ε_t are not, the aim is to estimate this model's parameter $\theta = \{\theta_1, RC, \rho\}$, given the data $\{x_t, i_t\}_{t=0}^T$, by maximum likelihood estimation.

[5] Since one can estimate the mileage transition process $Pr(x_{t+1} \,|\, i, x_t)$ – referred to as parameter θ_3 in the original model – independently from $\theta = \{\theta_1, RC, \rho\}$, and moreover, since it is exactly the same as in Rust (1987) (because it is not affected by the serial correlation in the unobserved state variables) we ignore this aspect of the bus engine replacement model in the remainder of this paper.

[6] Furthermore, we assume that $\varepsilon_0 = 0$; thus, ε_1 is distributed with density $q(\cdot)$.

[7] While the assumption that serial correlation is only present for regular maintenance utility shocks provides some computational simplification, one can also argue that it is easier to intrinsically motivate serial correlation for this case, because the errors are bus specific by construction; for example, one might think of a bus having some larger damage that persistently increases maintenance cost until the next general overhaul occurs.

3 Computation and Estimation

The following subsections develop the necessary numerical methods to estimate the bus engine replacement model of Rust (1987), with serially correlated unobserved state variables: First, we show how to approximate the solution to the dynamic problem, using numerical quadrature, interpolation over an adaptively refined grid, and solving a large non-linear system of equations. Then, we develop a method to decompose and approximate the integral over the unobserved state variables that appears in the likelihood function, and approximate it using highly efficient Gaussian quadrature. Finally, we describe a nested fixed point algorithm to obtain maximum likelihood estimates of the model parameters.

Note that this procedure is not specific to the model under consideration, but rather generic with respect to DDC models with $AR(n)$ type serial correlation.

3.1 The Expected Value Function

From (2) it is clear that in order to obtain the value function, we need to compute its conditional expectation. In fact, the computation of the likelihood function actually requires the expected value rather than the value itself (see Section 3.2). Thus, this section describes the necessary steps to numerically approximate the expected value as a function of all possible states:

$$EV_\theta(x,\varepsilon) = \sum_{x'} \int_{\varepsilon'} \max_{i\in\{0,1\}} \left\{ u_\theta(i,x') + \varepsilon'(i) + \beta EV_\theta(x',\varepsilon') \right\} Pr(\mathrm{d}\varepsilon' \mid \varepsilon, \theta)\, Pr(x' \mid x) \equiv T(EV_\theta)(x,\varepsilon) \tag{8}$$

Keeping the original time structure of the expectation (6) in mind, the expectation on the leftmost side of (8) is – strictly speaking – taken at time t, while the one on the right hand side (within the max operator) is taken at time $t+1$. But since the value function and its expectation are time invariant, given state (x,ε), the same unknown function EV_θ appears on both, the left and the right side of the equation. Therefore, EV_θ is the solution to the functional equation

$$EV_\theta(x,\varepsilon) = T(EV_\theta)(x,\varepsilon) \tag{9}$$

and thus a fixed point of the non-linear operator T. Moreover, since T can be shown to have the contraction mapping property (Rust, 1988), this fixed point is unique and attractive.

The numerical approximation of (8) involves three main computational tasks:[8] First, we need to approximate the integral in (8) by numerical quadrature. Second, we have to approximate the continuous function EV_θ by a finite number of parameters, for example by interpolation. Finally, since EV_θ is only defined implicitly as a fixed point of T – and we therefore cannot evaluate it directly – we need to solve for the parameters of the function approximation by solving a non-linear system (or fixed point iteration).

Numerical integration. In contrast to the case of extreme value type I iid distributed unobserved state variables, no closed form solution to the integral (8) exists; thus, we have to approximate it by numerical quadrature. A variety of methods for multi-dimensional integration exists; see, for example, chapter 7 of Judd (1998) for an overview, or chapter 4 of Press et al. (2007) for an implementation oriented approach. Throughout the paper, we use Gaussian quadrature, which is known to be very efficient for the integration of functions that can be well approximated by a polynomial. While this condition is obviously violated for the value function (because of the kink potentially induced by the max-operator), one can still show Gaussian schemes to be convergent for any Riemann integrable function, and, moreover, they are reported to often outperform other widely used integration schemes, even in the presence of singularities;

[8] Generally, there is one more task, namely maximizing the utility and continuation value, in order to get the current value as a function of the states. However, since the choice set is discrete and small, we do the maximization by complete enumeration.

see Judd (1998) and the literature cited therein. Also, Stinebrickner (2000) successfully applied the Gaussian quadrature rules to expected value function approximation for DDCM with serial correlation.

The n-node Gaussian quadrature rule approximates

$$\int_a^b f(y)w(y)\mathrm{d}y \approx \sum_{i=1}^{n} \omega_i f(y_i) \tag{10}$$

where $w(y)$ is a non-negative weighting function with finite integral (including unity for $|a|, |b| < \infty$). The integration nodes y_i are the roots of the degree n polynomial of the family of polynomials that are mutually orthonormal with respect to weighting function $w(y)$.[9] The corresponding weights ω_i are chosen such that every polynomial of degree $2n-1$ is integrated *exactly*; for the corresponding formulas, see e.g. Kythe and Schäferkotter (2005). Since both, nodes and weights should be computed to high accuracy, they are often tabulated for some frequently used families of orthonormal polynomials.

When taking expectations of functions of continuous random variables, the integration problem (10) arises naturally, with the density function being used as weighting function $w(x)$. Obviously, this approach requires the availability of polynomials that are orthonormal with respect to the density function in use. For some distributions, these families are well known, such as the Hermite polynomials for normally distributed random variables. For most other distributions however, the necessary polynomials (and their roots) are unknown, and have to be computed first.[10] Alternatively, one can map the support of the corresponding density function to $[-1, 1]$ by a change of variable,[11] and approximate the resulting integral using the Gaussian rule based on Legendre polynomials, which are orthonormal with respect to the unity weighting function on $[-1, 1]$. Using this procedure, we found that expectations of extreme value distributed random variables can be approximated quite efficiently.

Directly approximating (8) by Gaussian quadrature has a potential caveat, since it would require to find polynomials that are orthonormal with respect the conditional probabilities, $Pr(\varepsilon' \,|\, \varepsilon)$, and thus different nodes and weights for each ε. Conseqently, we reformulate the integral in (8) in terms of the unconditional probability $q(\tilde{\varepsilon}'(i))$,

$$\begin{aligned}\int_{\tilde{\varepsilon}'(0)} \int_{\tilde{\varepsilon}'(1)} \max\big\{ & u(0,x') + \rho\varepsilon(0) + \tilde{\varepsilon}'(0) + \beta EV_\theta(x', (\rho\varepsilon(0) + \tilde{\varepsilon}'(0), \varepsilon'(1)), \\ & u(1,1) + \tilde{\varepsilon}'(1) + \beta EV_\theta(1, (0, \varepsilon'(1)))\big\}\, q(\mathrm{d}\tilde{\varepsilon}'(1)) q(\mathrm{d}\tilde{\varepsilon}'(0))\end{aligned} \tag{11}$$

and compute (or look up) one single set of nodes and weights for weighting function $q(\tilde{\varepsilon}'(i))$.[12]

Since the integration in (11) has dimension $N = 2$,[13] but Gaussian rules are per se one-dimensional, we use them extended to N dimensions by the product rule, which generalizes (10)

[9] A family of polynomials $\{\varphi_k(y)\}_{k=0}^{\infty}$ with inner product $\langle \varphi_k, \varphi_l \rangle = \int_a^b \varphi_k(y)\varphi_l(y)w(y)\mathrm{d}y$ is orthonormal with respect to weighting function $w(y)$ on $[a, b]$, if $\langle \varphi_k, \varphi_l \rangle = 0\ \forall k, l : k \neq l$, and $\langle \varphi_k, \varphi_k \rangle = 1\ \forall k$.

[10] One can for example use the Stieltjes' procedure (see e. g. Press et al., 2007), but the method should be used with caution, since it is prone to roundoff errors, and, moreover, contains difficult numerical integration problems itself.

[11] For example, if the inverse of the cumulative distribution of a distribution with density $w(y)$, $W^{-1}(y)$, exists, one can apply the following change of variables: $\int_{-\infty}^{+\infty} f(y)w(y) = \int_0^1 f(W^{-1}(y))$.

[12] Equation (11) silently assumes that after a replacement, the series of serially correlated unobserved states is reset to its mean, 0. Thus, ε in the first period after an engine replacement is distributed according to density $q(\cdot)$ again.

[13] The dimension of the integration over the unobserved state variable in DDCM is usually $(N-1)$-dimensional, because the decisions of the agents in the model are driven by utility differences rather than levels. In this case however, since we assume that serial correlation is only present in one dimension of the error, the reformulation of the model in terms of the differences of errors does not reduce dimensionality. Thus, the integration must be carried out over all the N dimensions.

to N dimensions by

$$\int_{[a,b]^N} f(y^1,\ldots,y^N)\prod_{i=1}^{N} w_i(y^i)\mathrm{d}(y^1,\ldots,y^N) \approx \sum_{i_1=1}^{N}\cdots\sum_{i_N=1}^{N} f(y^1_{i_1},\ldots,y^N_{i_N})\prod_{j=1}^{N}\omega^j_{i_j} \tag{12}$$

where $f:\mathbb{R}^N\mapsto\mathbb{R}$, $w_i:\mathbb{R}\mapsto\mathbb{R}$ is the weighting function for dimension i, and y^i_j and ω^i_j are the nodes and weights of the corresponding one-dimensional Gaussian rule (indexed by j), applied to dimension i.[14]

Function approximation. Generally, the expected value function is a continuous function of ε, and we need to approximate it as such, but by a finite number of parameters only. Assume for the moment that we can evaluate an unknown function $f(y)$ at arbitrary points, and that $f(y)$ is deterministic.[15] Then, we can choose a set of nodes $y_i\in[a,b]$, and construct an interpolating function $\hat{f}(y)$, such that $f(y_i)=\hat{f}(y_i)\ \forall y_i$.[16] Obviously, we want to choose $\hat{f}(y)$ such that $|f(y)-\hat{f}(y)|$ is "small everywhere", not just at the interpolation nodes y_i. More formally, we want to control the interpolation error $\sup_{y\in[a,b]}|f(y)-\hat{f}(y)|$. However, the problem of finding an interpolation scheme and nodes, that deliver good approximation quality, is problem specific.[17]

A general, but computationally expensive approach to node choice are adaptive procedures: Given some interpolant $\hat{f}^{(h)}(y)$, we evaluate the quality of approximation, $|f(y)-\hat{f}^{(h)}(y)|$, at different values of the argument (different from y_i), and we insert new nodes where the approximation quality is poor; then, we construct a new interpolant $\hat{f}^{(h+1)}(y)$ on the set union of old and new nodes. This procedure is iterated until some convergence criterion is met. Adaptive methods are particularly well suited for functions with "difficult" shape properties, for example functions with greatly varying curvature, kinks, or discontinuities, and to explicitly control the approximation error. For the actual interpolation over such a grid, one can for example use piecewise methods, such as splines.[18]

Since we want to have direct control over the error of the approximation of EV_θ, we choose an adaptive approximation method; in particular, we want to assure uniform approximation quality for different values of θ, in order to compute the corresponding likelihood function values to high accuracy. Therefore, we employ the method of Grüne and Semmler (2004), which repeatedly refines an interpolation grid, until a global approximation error criterion is met. At this point, it is important to note that we cannot directly evaluate the true (but unknown) expected value function EV_θ, because it is only implicitly defined by (9). Fortunately, to discuss this grid adaption method, it is sufficient to assume that the method is supplied with an approximation $\widehat{EV}^{(h)}_\theta(\,\cdot\,;a)$ from the previous iteration of the adaption process, which is now explicitly parametrized by the finite-dimensional vector $a\in\mathbb{R}^A$. Let $\Gamma^{(h)}_\theta$ be the grid at the

[14] Note that in order to use the product rule (12) to compute expectations, the dimensions of the random variable must be mutually independent. For more general multivariate distributions, see, for example, Jäckel (2005).

[15] A function is called deterministic, if it is not subject to noise, thus, if the variance of the function value at a given argument is zero on its whole domain. For example, $f:\mathbb{R}\mapsto\mathbb{R}$ is deterministic if $Var(f(y))=0,\ \forall y\in\mathbb{R}$.

[16] Unlike interpolation, perturbation methods such as Taylor series approximation only require the evaluation of the unknown function for one particular value. However, for Taylor series approximation to obtain accurate results, it is often necessary (but not sufficient) to also evaluate higher order derivatives of the function. On the other extreme, regression methods evaluate the function at much more points than interpolation methods usually use, in order to "filter out" information that is not relevant for the approximation, such as noise. Obviously, the interpolation property does not hold for functions obtained from regression.

[17] While the two main steps of choosing the nodes, as well as choosing and parametrizing the interpolation function might seem independent, the performance of a particular interpolation scheme usually depends on a reasonable combination of the two; a well known example is the approximation by Chebyshev polynomials, for which one can show favorable convergence properties, if the nodes are chosen to be the roots of the corresponding polynomials.

[18] For arbitrary choices of interpolation nodes, polynomial approximation schemes do not necessarily converge if more interpolation nodes are added; this property is known as the Runge phenomenon, and makes it particularly difficult to use polynomial approximation in conjunction with adaptive node choice.

beginning of iteration h. For each cell[19] c_l of grid $\Gamma_\theta^{(h)}$, we approximate the solution to the following optimization problem[20]

$$\eta_l = \max_{\varepsilon \in c_l} |\widehat{EV}_\theta^{(h)}(x, \varepsilon; a) - T(\widehat{EV}_\theta^{(h)})(x, \varepsilon; a)| \tag{13}$$

Then, Grüne (1997) showed that the maximum error over all cells, $\eta = \max_l\{\eta_l\}$, defines an approximation error bound by

$$\max_{x \in X, \varepsilon \in R^N} |EV_\theta(x, \varepsilon) - \widehat{EV}_\theta^{(h)}(x, \varepsilon; a)| \leq \eta \frac{1}{1-\beta} \tag{14}$$

where EV_θ represents the true (but unknown) expected value function. The method of Grüne and Semmler (2004) inserts new nodes into those cells c_l where the corresponding error η_l is larger than some threshold. Finally, we construct new interpolant $\widehat{EV}_\theta^{(h+1)}(\,\cdot\,; a)$ on the refined grid $\Gamma_\theta^{(h+1)}$. (In order to parametrize it, we need to solve for the fixed point (9), which we discuss shortly.) This procedure is repeated, until the maximum (global) approximation error $\eta(1-\beta)^{-1}$ is smaller than the desired approximation error, $\bar{\eta}$.

One particular advantage of the method of Grüne and Semmler (2004) is that it not only allows for refinement, but easily extends to grid coarsening, by identifying and removing nodes that do not increase approximation accuracy. Combining coarsening and refinement, we can construct a grid *updating* procedure, which can be integrated with a nested fixed point algorithm (NFXP). In NFXP, the likelihood maximization ("outer loop") repeatedly feeds different values of θ into the expected value function approximation ("inner loop"); thus, rather than building up from scratch an interpolant for each new value of $\theta^{(k+1)}$, it can be obtained from updating an interpolant that has previously been build for some other value $\theta^{(k)}$ (see Section 3.3 below).

Note that due to the fact that serial correlation is only allowed in $\varepsilon(0)$, $EV_\theta(x, \varepsilon)$ is constant in $\varepsilon(1)$. Consequently, we only need to approximate it as a one-dimensional function of $\varepsilon(0)$. Therefore, we can use piecewise linear interpolation to construct $\widehat{EV}_\theta$.[21] However, the methodology generalizes to higher dimensions by replacing PLI with multidimensional interpolation.

Finally note that, since in this formulation of the model, mileage has been discretized, we need to approximate EV_θ as a separate continuous function of ε for each mileage state $x \in X$ simultaneously; thus, $\widehat{EV}_\theta(\,\cdot\,; a)$ is really a set of interpolants. If, in contrast, mileage would enter the model as a continuous variable, $\widehat{EV}_\theta(\,\cdot\,; a)$ would rather be a single 2-dimensional interpolant. However, discrete mileage is necessary to nest the original model without serial correlation as a special case.

Non-linear system. The last few paragraphs discussed the choice of a function approximation scheme and interpolation grid creation, but left out how to actually evaluate the unknown function EV_θ, which is only implicitly defined as the fixed point of T. While this fixed point is generally a continuous function, its substitution by an approximating interpolant $\widehat{EV}_\theta(\,\cdot\,; a)$ simplifies the problem to a non-linear system of D equations in A unknowns,

$$\widehat{EV}_\theta(x, \varepsilon; a) = T(\widehat{EV}_\theta)(x, \varepsilon; a) \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{15}$$

where D is the number of elements in Γ_θ, and thus each $(x, \varepsilon) \in \Gamma_\theta$ defines one equation of (15), and the parameters a of the interpolant are the variables. From the parameter vector a^*

[19] In this context, cell c_i of a n-dimensional grid Γ is defined as the hypercube spanned by $\left\{y_j \in \Gamma : y_i^k \leq y_j^k \leq \min_l\{y_l^k : y_i^k < y_l^k\}, k = 1, \ldots, n\right\}$, where y^k is the kth element (dimension) of the vector y.

[20] Note that since the model is already discretized in terms of mileage state x, finding the maximum error within each each cell does not explicitly involve x; rather, one has to carry out the error estimation for all possible mileage states independently.

[21] While piecewise linear interpolation usually takes more nodes to produce sufficiently smooth approximations compared to higher order methods – if the function to be approximated is really smooth – PLI showed better stability in the parametrization step of solving the non-linear system in our case. Also, the higher accuracy needs we imposed, the more PLI and higher order methods became comparable in performance.

that solves (15), we can directly construct the interpolant $\widehat{EV}_\theta(\cdot\,; a^*)$. This procedure is known as collocation, which is a particular variant of a projection method for the approximation of functions that are defined by functional equations; see Judd (1998), chapter 11. Finally, we compute the approximation error of $\widehat{EV}_\theta(\cdot\,; a^*)$ as defined by (14); if it is sufficiently small (smaller than $\bar{\eta}$), we accept our approximation of EV_θ, otherwise, we refine the interpolation grid Γ_θ, and solve (15) for the new grid.[22]

Since for piecewise linear interpolation, the parameters of the approximation correspond to the function values and the slopes at the grid points (and the slopes can be derived from the neighboring grid points), the system (15) is square, so $D = A$. Moreover, it is actually still a fixed point problem in the parameter vector a, because a is identical to the function values on the grid, and therefore identical to the left hand side of (15):

$$a = T'(a; x, \varepsilon) \equiv T(\widehat{EV}_\theta)(x, \varepsilon; a) \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{16}$$

Consequently, one could use fixed point iteration schemes to get "close" to the solution (16) quite fast.[23] However, using fixed point iteration to solve for the approximation of EV_θ does not deliver the accuracy needed to guarantee convergence in the outer loop of likelihood maximization in our case (see Dubé et al., 2012, for this issue).

Consequently, similar to Rust (1987), we use methods that directly solve the non-linear system

$$\widehat{EV}_\theta(x, \varepsilon; a) - T(\widehat{EV}_\theta)(x, \varepsilon; a) = 0 \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{17}$$

to high accuracy. Given the accuracy needs of our application, Newton methods are particularly interesting, because they show quadratic convergence close to the solution under some conditions.[24] However, these methods require the evaluation of the Jacobian matrix J of the non-linear system (17), which is generally of size D^2, and thus can be prohibitively expensive to compute for large systems. In particular, given an adaptively refined grid, the size of J can become an issue since the number of equations of (17) is defined by the number of nodes in Γ_θ, and thus the system grows larger as the grid is refined. However, analogously to the original model, if the Markov transition matrix of the discrete states is sparse, also J is sparse; thus, using Newton (or quasi-Newton) methods can still be feasible, because the number of non-zero elements in the Jacobian grows much slower in the number of grid nodes.[25] Figure 1 illustrates the sparseness pattern of our problem.

To numerically solve the fixed point problem (9), we use the "ipopt" package (Wächter and Biegler, 2005), in conjuction with the "pardiso" sparse linear solver (Schenk and Gärtner, 2004).

Figure 2 plots an example of the expected value function, where each of the black lines represents the expected value as a function of $\varepsilon(0)$, for a particular value x. We want to

[22] Note that in contrast to interpolation of explicitly defined functions, the interpolation property with respect to the true expected value function, $\widehat{EV}_\theta(x, \varepsilon; a) = EV_\theta(x, \varepsilon),\ \forall (x, \varepsilon) \in \Gamma_\theta$, generally only holds as the approximation error $\eta(1-\beta)^{-1}$ goes to zero, because for a positive approximation error, $T(EV_\theta)(x, \varepsilon) = EV_\theta(x, \varepsilon) \neq T(\widehat{EV}_\theta)(x, \varepsilon; a),\ (x, \varepsilon) \in \Gamma_\theta$.

[23] Note that in principle, one can apply the fixed point formulation (16) to every interpolation scheme that uniquely maps the function values on the grid into the parameters, which is actually true for many interpolation schemes. For example using a spline, one can implement the fixed point representation in a two step procedure, which first does a Newton iteration on the function values, and then fits a new spline over the grid (as part of operator T'). However, numerically, it can be more stable to add the system of equations that determines the parameters of the spline to (15), and solve the system in the parameters directly.

[24] Suppose for $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, a solution y^* to the system $f(y^*) = 0$ exists, the Jacobian function $J : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ is Lipschitz continuous, and the Jacobian matrix at the solution, $J_f(y^*)$, is non-singular. Then, if $y^{(0)}$ is sufficiently close to the solution y^*, the residual decays quadratically for each Newton iteration, thus $\exists K > 0 : \|y^{(k+1)} - y^*\| \leq K\|y^{(k)} - y^*\|^2$. Loosely speaking, close to the solution, the number of correct digits of the result roughly doubles in every Newton step.

[25] The complexity of evaluating the Jacobian is still quadratic in the number of equations, but with a small multiplicative constant $c \ll 1$, which is decaying in the degree of sparseness. Thus, for sufficiently sparse transition matrices, it is well feasible to evaluate the Jacobian, even more since one can evaluate its elements perfectly parallel.

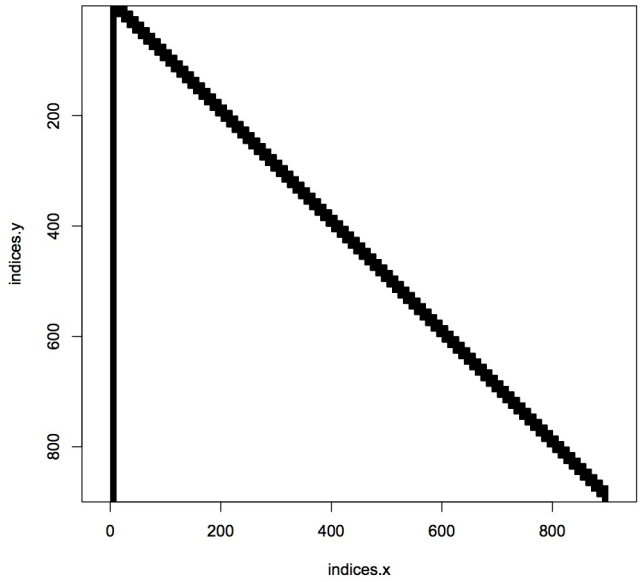

Figure 1: Sparseness pattern of the Jacobian of the non-linear system (17).

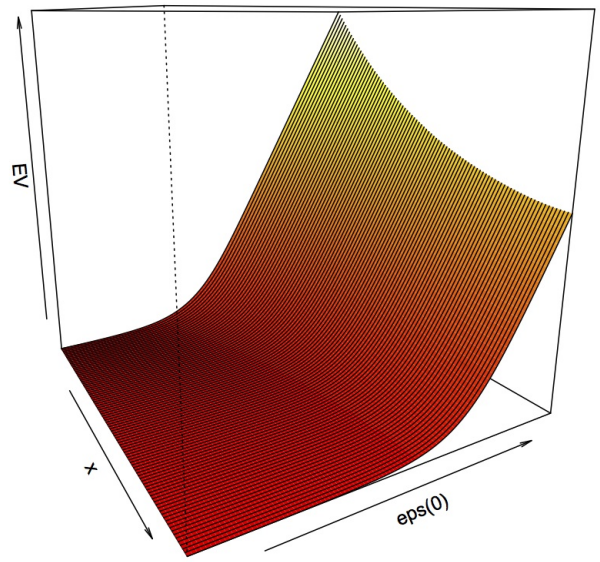

Figure 2: The expected value function $EV_\theta(x, \varepsilon)$ for $\rho = 0.6$, $RC = 14$, $\theta_1 = 2$, $\tilde{\varepsilon}(i) \sim$ EV1 iid.

emphasize again that the procedure to compute an approximation of $EV_\theta(x, \varepsilon)$ as presented in this section easily generalizes to other models, with arbitrary number of decisions N, and serial correlation in all dimensions of the unobserved state variables, by choosing a multi-dimensional interpolation scheme.

3.2 The Likelihood Function

In this section, we derive the likelihood function for the bus engine replacement model with serially correlated unobserved state variables, and formulate it such that the dimensionality of the numerical integration only depends on the number of choices N, but not on the time horizon of the observation, T. In a second step, we provide a numerical procedure to solve this formulation, using standard deterministic quadrature rules to high accuracy. It is important to note that this reformulation is not specific to the Rust (1987) model, but generically applies to DDCM with $AR(n)$ type serial correlation, even one allows for serial correlation in the errors for all choices.

The likelihood function of one individual bus derives as follows:

$$L(\,\theta\,|\,\{x_t, i_t\}_{t=0}^T) = \int_{\varepsilon_0,\ldots,\varepsilon_T} \cdots \int Pr(\{x_t, i_t, \varepsilon_t\}_{t=0}^T \,|\, \theta)\; \mathrm{d}\varepsilon_0 \ldots \mathrm{d}\varepsilon_T \tag{18}$$

The likelihood function of the full panel computes as the product of the likelihood functions of the individual buses, since the state variables are assumed to be independently distributed across buses. Incorporating the assumption that all state transitions are Markov, we can factorize the probability of observing a particular time series as

$$Pr(\{x_t, i_t, \varepsilon_t\}_{t=0}^T \,|\, \theta) = \prod_{t=1}^{T} Pr(x_t, i_t, \varepsilon_t \,|\, x_{t-1}, i_{t-1}, \varepsilon_{t-1}, \theta). \tag{19}$$

We can further decompose the joint transition probability in (19), using the fact that, given x_t and ε_t, i_t is independent of i_{t-1}, ε_{t-1} and x_{t-1},[26] as well as incorporating assumption (5):

$$Pr(x_t, i_t, \varepsilon_t \,|\, x_{t-1}, i_{t-1}, \varepsilon_{t-1}, \theta) = Pr(i_t \,|\, x_t, \varepsilon_t, \theta) Pr(\varepsilon_t \,|\, i_{t-1}, \varepsilon_{t-1}, \theta) Pr(x_t \,|\, x_{t-1}, i_{t-1}) \tag{20}$$

For notational simplicity, define

$$m_{it} \equiv u_\theta(i, x_t) + \beta \mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t]. \tag{21}$$

While $Pr(\varepsilon_t \,|\, i_{t-1}, \varepsilon_{t-1}, \theta)$ is determined by (7) and $Pr(x_t \,|\, x_{t-1}, i_{t-1})$ is estimated independently (and therefore omitted from now on)[27], the conditional decision probability $Pr(i_t \,|\, x_t, \varepsilon_t, \theta)$ is given by

$$Pr(i_t = 1 \,|\, x_t, \varepsilon_t(0), \varepsilon_t(1), \theta) = \mathbb{1}(m_{1t} + \varepsilon_t(1) > m_{0t} + \varepsilon_t(0)) \tag{22}$$

where $\mathbb{1}(\cdot)$ is the index function that is equal to one if its argument is true, and zero otherwise; note that the conditional decision probabilities are actually degenerate, because, loosely speaking, there is no randomness left, given ε_t.

Finally, exploiting the Markov structure for the integration, and dropping parameter dependence for better readability, we can write the likelihood function (18) as

$$L(\,\theta\,|\,\{x_t, i_t\}_{t=0}^T) = \int_{\varepsilon_0} \ldots \int_{\varepsilon_{T-1}} \ldots \int_{\varepsilon_T} Pr(i_T \,|\, x_T, \varepsilon_T) Pr(\varepsilon_T \,|\, i_{T-1}, \varepsilon_{T-1})\; \mathrm{d}\varepsilon_0 \ldots \mathrm{d}\varepsilon_{T-1} \mathrm{d}\varepsilon_T. \tag{23}$$

[26] This fact follows from the definition of the value function (2).

[27] Since one can estimate the mileage transition probabilities separately, they only add a multiplicative constant to the likelihood function of $\theta = \{\theta_1, RC, \rho\}$. Thus, we omit the corresponding term of the likelihood function (and one should so in the actual maximization for scaling reasons).

To numerically approximate (23), define the function

$$g_t(\varepsilon) = \begin{cases} 1 & t > T \\ \int_{\varepsilon'} Pr(i_{t+1} \,|\, x_{t+1}, \varepsilon') Pr(\varepsilon' \,|\, i_t, \varepsilon) g_{t+1}(\varepsilon') \, \mathrm{d}\varepsilon' & \text{otherwise} \end{cases} \tag{24}$$

Now, given $g_{t+1}(\varepsilon)$, we can numerically approximate the function $g_t(\varepsilon)$ using both, numerical integration and function approximation. Since $g_t(\varepsilon)$ is known to be unity for $t > T$, we can use backward iteration starting from $g_T(\varepsilon)$, to solve for $g_0(\varepsilon)$, which is the approximation of the likelihood function $L(\,\theta \,|\, \cdot)$. Note that this procedure is analogous to solving for the value function of a finite horizon, discrete time dynamic programming problem by backward iteration. Algorithm 1 gives a formal description of the procedure.[28]

Algorithm 1 Backward iterative computation of the likelihood function (23).

```
discretize support of ε → Γ ∈ ℝ^D
initialize interpolant ĝ(·) with nodes {ĝ_e}_{e∈Γ} ∈ ℝ^D to unity
for t ∈ T, ..., 1 do
    for e ∈ Γ do
        ĝ_e ← approximate ∫_{ε'} Pr(i_{t+1} | x_{t+1}, ε') Pr(ε' | i_t, ε) ĝ(ε') dε'
    end for
    ĝ(·) ← construct interpolant with nodes {ĝ_e}_{e∈Γ}
end for
```

Note that each integral over ε_t is generally still N-dimensional. Thus, the procedure decomposes the $T \cdot N$-dimensional integral of (18) to a N-dimensional integration that is repeated $D \cdot T$ times, where D is the number of nodes used for function approximation of g_t. Since the computational complexity of deterministic numerical integration is generally exponential in the number of dimensions, this reduction is highly desirable even for large D, because it enters complexity of the overall algorithm linearly[29]

$$O(\exp(T \cdot N)) \gg O(D \cdot T \exp(N)) \tag{25}$$

Given that serial correlation is only allowed in some dimensions, but not all, we can potentially replace parts of integral in (24) by a closed form solution; this is particularly the case if the cumulative distribution of those unobserved state variables that are not serially correlated does have a closed form. Recall that the integration over ε_t really N-dimensional, thus 2-dimensional in the model under consideration:

$$\int_{\varepsilon_t(0)} \int_{\varepsilon_t(1)} Pr(\varepsilon_t(0) \,|\, i_{t-1}, \varepsilon_{t-1}(0)) \; Pr(\varepsilon_t(1)) \; Pr(i_t \,|\, x_t, \varepsilon_t(0), \varepsilon_t(1)) \; \mathrm{d}\varepsilon_t(1) \mathrm{d}\varepsilon_t(0) \tag{26}$$

Using (22), we can write the integral over $\varepsilon_t(1)$ in terms of its cumulative distribution function F,

$$\begin{aligned} &\int_{-\infty}^{\infty} \mathbb{1}(\varepsilon_t(1) > m_{0t} - m_{1t} + \varepsilon_t(0)) \; Pr(\varepsilon_t(1)) \; \mathrm{d}\varepsilon_t(1) \\ &= \int_{m_{0t} - m_{1t} + \varepsilon_t(0)}^{\infty} Pr(\varepsilon_t(1)) \; \mathrm{d}\varepsilon_t(1) = 1 - F(m_{0t} - m_{1t} + \varepsilon_t(0)) \end{aligned} \tag{27}$$

[28] Algorithm 1 is generic with respect to both, the numerical integration scheme as well as the function approximation schemes, as long as the latter depends on function evaluations only.

[29] In the context of complexity analysis of algorithms, the $O(f(y))$ notation reads as follows: There exists a constant $K > 0$ such that the number of iterations needed for an algorithm to complete a task of size y is bounded by $K \cdot f(y)$.

which no longer involves numerical quadrature if an analytical formula for F exists.

For the actual computations we use Gaussian quadrature as outlined in the previous section (in the context of expected value function approximation). Note that while we write all integrals in this section as integrals over ε for simplicity, we have to reformulate them in terms of $\tilde{\varepsilon}$ by a linear change of variables, in order to approximate them by Gaussian quadrature (see Section 3.1). Also, for numerical reasons, we chose a slightly different change of variables to map the integration domain from $[-\infty, \infty]$ to $[-1, 1]$, (see Judd, 1998, p. 204). Furthermore, we use Akima splines (Akima, 1970) to approximate the integral over ε_t as a function of ε_{t-1}.

3.3 Likelihood Function Maximization

The maximum likelihood estimate of θ, given data $\{x_t, i_t\}_{t=0}^T$, is the solution to the following optimization problem:

$$\begin{aligned} \max_{\theta} \; & L(\,\theta \,|\, \{x_t, i_t\}_{t=0}^T) \\ \text{s.t. } & EV_\theta(x, \varepsilon) = T(EV_\theta)(x, \varepsilon) \end{aligned} \tag{28}$$

While there exist methods that directly solve the constrained problem (28), namely the mathematical programming with equilibrium constraints (MPEC) approach, that Su and Judd (2012) successfully applied to DDCM, and in particular to the bus engine replacement model of Rust (1987), we use the well known nested fixed point (NFXP) approach by Rust (1988):[30] Instead of solving the constrained optimization problem (28), the likelihood maximization is performed as a two step procedure. First, given a parameter guess $\theta^{(k)}$, one computes the expected value function $EV_{\theta^{(k)}}$ as a fixed point of operator T. Second, one evaluates the likelihood function for $\theta^{(k)}$, using the approximation of $EV_{\theta^{(k)}}$ just obtained before. The optimization algorithm then constructs a new parameter guess $\theta^{(k+1)}$. The procedure starts again by approximating $EV_{\theta^{(k+1)}}$, and is iterated until convergence of the maximization algorithm.[31] Thus, the constrained optimization problem (28) is actually transformed into an unconstrained problem.

Recall that the interpolation grid $\Gamma_{\theta^{(k)}}$, over which the corresponding approximating interpolant $\widehat{EV}_{\theta^{(k)}}(\,\cdot\,; a)$ satisfies some error bound $\bar{\eta}$, depends on $\theta^{(k)}$. Thus, each step of the maximization routine, from $\theta^{(k)}$ to $\theta^{(k+1)}$, requires to iteratively update $\Gamma_{\theta^{(k)}}$ to $\Gamma_{\theta^{(k+1)}}$, until the maximum approximation error of $\widehat{EV}_{\theta^{(k)}}(\,\cdot\,; a)$ is bounded by $\bar{\eta}$ again; this procedure ensures that for each likelihood function evaluation, the approximation error of the corresponding expected value function is controlled.[32]

Algorithm 2 summarizes the nested fixed point algorithm to solve (28).

For the model under consideration, the maximization of the likelihood function is an unconstrained non-linear optimization problem with three free parameters. We use a quasi-Newton

[30] The MPEC approach to DDCM estimation of Su and Judd (2012) "combines" the solution of the fixed point and the maximization of the likelihood, by solving the original constraint formulation of the likelihood maximization problem (28). This procedure is generally more efficient than the NFXP approach, because it does not require to solve the fixed point equation (9) for each parameter guess, even if it is far away from the solution; rather, it imposes the fixed point condition to hold only at the solution. However, directly integrating MPEC with adaptive interpolation grids creates two potential problems: First, adding a grid node corresponds to adding a constraint to the optimization problem, while the optimization algorithm runs. Second, adaptive methods usually require the approximation of an iteration to be completed in order to compute the approximation quality for the insertion decision, which in our case is not possible until (9) has been solved, which in turn contradicts the MPEC idea.

[31] Since the fixed point of T is usually obtained using an iterative method, solving the dynamic problem is often referred to as the "inner loop" in this context, while the maximization procedure is referred to as the "outer loop".

[32] Controlling the maximum approximation error does not imply that it is constant over the maximization procedure. Rather, we choose $\bar{\eta}^{(k)}$ to be decreasing in the iterations of the optimizer, in order to compute the fixed point to lower accuracy far away from the solution, but to high accuracy close to it.

Algorithm 2 Nested fixed point algorithm with adaptive grid updating.

initialize θ, Γ_θ, a
while $\nabla L(\theta) \neq 0$ **do**
  while $\eta(1-\beta)^{-1} > \bar{\eta}$ **do**
    solve $\widehat{EV}_\theta(x,\varepsilon;a) = T(\widehat{EV}_\theta)(x,\varepsilon;a) \quad \forall (x,\varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A$
    update Γ_θ (coarsening and refinement)
  end while
  evaluate $L(\theta), \nabla L(\theta)$
  compute next θ
end while

trust-region method, provided by the R-package "trustOptim" (Braun, 2012); for a comprehensive description of this method, see e.g. Nocedal and Wright (2006).[33]

4 Estimation Results

The original dataset of Rust (1987) consists of monthly odometer readings and engine replacement decisions for a fleet of 162 buses, subdivided into 8 groups depending on their manufacturer and model. Since buses are heterogeneous among groups, it is common to create different subsamples to estimate the parameters of model (1); we follow the literature by estimating three subsamples separately, consisting of groups $\{1,2,3\}$, $\{1,2,3,4\}$, and $\{4\}$. Table 1 shows the size of the panel for each group under consideration.[34]

bus group	number of buses	observation horizon (months)	total number of observations	number of replacements
1	15	25	360	0
2	4	49	192	0
3	48	70	3312	27
4	37	117	4292	33
total	104		8156	60

Table 1: Number of buses, observation time horizon in months, total number of observations, and number of observed engine replacements for each bus group.

As in Rust (1987), we discretize mileage in "bins" of 5,000 miles each.[35] The highest possible mileage state is 90 (which corresponds to 450,000 miles),[36] formally $x \in X = \{1,\ldots,90\}$. We assume the mileage transition to follow a Markov process (conditional on the replacement decision), for which we estimate the parameters independently. We parametrize the discount factor as in the original paper by $\beta = 0.9999$.

[33] Note that for the trust-region method to work efficiently, proper scaling must be applied to the objective function and the parameters. Also, due to the potentially high computational noise in evaluating the likelihood function, the step length of the finite difference approximations of the gradients should be chosen with care. For a discussion of these issues, see e.g. Gill et al. (1981).

[34] The number of observations is equal to the time horizon in months, minus one (because there is no mileage transition in the first observation of each bus), times the number of buses.

[35] By discretizing into bins of 5,000 miles we mean that the original mileage $\tilde{x}$ transforms into a mileage state $x = \lceil \tilde{x}/5,000 \rceil$, with the ceiling function $\lceil \tilde{y} \rceil = \min\{y \in \mathbb{N} : y \geq \tilde{y}\}$.

[36] If a bus ever reaches the maximum mileage state, we assume it to stay there until engine replacement. Although no bus in any of our subsamples ever reaches the maximum mileage state, it still has relevance for the solution of the dynamic problem of the agent, who takes this possibility into account when solving his infinite horizon dynamic optimization problem.

Before presenting the results of the estimation, we verify the estimation procedure presented in Section 3: First, Table 2 presents a partial reproduction of Table IX of Rust (1987), without serial correlation, but still numerically integrating both expected value and the likelihood function. Second, Table 3 presents the results of the estimation using artificial datasets with serial correlation. Since we know the parameters used for the simulation, we can directly test the ability of the algorithm to recover the original parameters; we do this verification for the density $q(\tilde{\varepsilon})$ being EV1, as well as normal. Observe that the procedure is able to recover the true parameters to high accuracy.

	Rust (1987)	estimated
RC	9.7558	9.7557
θ_1	2.6275	2.6274
ρ	–	–
L	-6055.250	-6055.250
$\|\|\nabla L\|\|$		1e-9

Table 2: Replication of Table IX of Rust (1987) for bus groups 1-4; L is the value of the log-likelihood function at the solution; $||\nabla L||$ is the order of magnitude of the norm of the gradient of the log-likelihood function at the solution; $\beta = .9999$.

	true	estimated	
		$EV1$	$N(0,1)$
RC	14.0000	13.9959	14.0325
θ_1	2.0000	2.0390	2.0464
ρ	0.6000	0.5997	0.5864
$\|\|\nabla L\|\|$		1e-5	1e-5

Table 3: Estimation of artificially generated datasets, for densities $q(\tilde{\varepsilon})$ being $EV1$ and standard normal $N(0,1)$; $||\nabla L||$ is the order of magnitude of the norm of the gradient of the log-likelihood function at the solution; $\beta = .9999$.

Tables 4 and 5 finally present the estimation results with serial correlation, using the original dataset of Rust (1987), again for both, $EV1$ (Table 4) and normally distributed $\tilde{\varepsilon}$ (Table 5). Note that in the $EV1$ case, while the parameter estimates in the presence of serial correlation are substantially different from the estimates without serial correlation, the ratio of engine replacement cost to the regular maintenance cost parameter is relatively stable; thus, the trade-off for the decision maker has not changed much quantitatively. Performing a likelihood ratio test to compute the statistical significance of the quantitative changes induced by the introduction of serial correlation, only on the largest subsample of the dataset (bus groups 1–4) we can reject the hypothesis of no serial correlation at reasonable significance levels. In case of normally distributed $\tilde{\varepsilon}$, both the parameter values and their ratios change substantially; however, we cannot reject the hypothesis of no serial correlation at a reasonable significance level for this dataset.[37]

[37]The failure to rejecting the no serial correlation hypotheses might well be due to a relatively small dataset; for example, the biggest subsample of the dataset used in this context (bus groups 1–4) only contains 60 replacement decisions.

	Bus Groups 1-3		Bus Groups 1-4		Bus Group 4	
RC	11.8270	25.0000	9.7557	26.4972	10.0740	22.4464
θ_1	4.6724	9.8347	2.6274	7.2392	2.2927	4.9162
RC/θ_1	2.5313	2.5420	3.7130	3.6602	4.3793	4.5658
ρ	–	0.6894	–	0.7366	–	0.7045
L	-2708.335	-2707.765	-6055.250	-6053.341	-3304.158	-3303.914
$\|\|\nabla L\|\|$	1e-7	1e-6	1e-9	1e-5	1e-5	1e-6
p (LR)		0.2854		0.0507		0.4848

Table 4: Estimation results for the original dataset, density $q(\tilde{\varepsilon})$ being $EV1$. L is the value of the log-likelihood function at the solution, $||\nabla L||$ is the order of magnitude of the norm of the gradient of the log-likelihood function at the solution; p (LR) is the p-value of the likelihood ratio test with $H_0 : \rho = 0$; $\beta = .9999$.

	Bus Groups 1-3		Bus Groups 1-4		Bus Group 4	
RC	7.0870	13.9130	6.0047	18.4240	6.0753	11.5717
θ_1	2.4586	5.4257	1.4011	5.1150	1.1829	2.4626
RC/θ_1	2.8826	2.5643	4.2857	3.6020	5.1359	4.6990
ρ	–	0.5230	–	0.6623	–	0.5155
L	-2707.877	-2707.820	-6054.084	-6053.685	-3303.919	-3303.901
$\|\|\nabla L\|\|$	1e-5	1e-5	1e-6	1e-5	1e-6	1e-5
p (LR)		0.7354		0.3713		0.8503

Table 5: Estimation results for the original dataset, density $q(\tilde{\varepsilon})$ being $N(0, 1)$. L is the value of the log-likelihood function at the solution, $||\nabla L||$ is the order of magnitude of the norm of the gradient of the log-likelihood function at the solution; p (LR) is the p-value of the likelihood ratio test with $H_0 : \rho = 0$; $\beta = .9999$.

5 Conclusion

This paper developed a method to efficiently estimate dynamic discrete choice models in the presence of serial correlation in the unobserved state variables. First, to approximate the expected value function of the underlying dynamic problem, we use Gaussian quadrature, interpolation over an adaptively refined grid, and solve a potentially large non-linear system. Second, to evaluate the likelihood function, we decompose the integral over the unobserved state variables in the likelihood function into a series of lower dimensional integrals, and successively approximate them using Gaussian quadrature rules. Finally, we solve solve maximum likelihood problem using a nested fixed point algorithm.

After verifying the algorithm's ability to recover the parameters using an artificial data set, we apply this method to the bus engine replacement model of Rust (1987), and we find significant serial correlation for some of the subsamples. Also, the parameter estimates vary substantially, compared to the case of serially uncorrelated errors. We want to emphasize again that the method presented in this paper is not limited to the bus engine replacement model, but is generic to DDCM models with $AR(n)$ type serially correlated errors.

References

Aguirregabiria, V. and Mira, P. (2010). Dynamic Discrete Choice Structural Models: A Survey. *Journal of Econometrics*, 156(1):38–67.

Akima, H. (1970). A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the ACM*, 17(4):589–602.

Arcidiacono, P. and Ellickson, P. B. (2011). Practical Methods for Estimation of Dynamic Discrete Choice Models. *Annual Review of Economics*, 3(1):363–394.

Arcidiacono, P. and Miller, R. A. (2011). Conditional Choice Probability Estimation of Dynamic Discrete Choice Models with Unobserved Heterogeneity. *Econometrica: Journal of the Econometric Society*, 79(6):1823–1867.

Braun, M. (2012). *trustOptim: Trust Region Nonlinear Optimization, Efficient for Sparse Hessians.*

Cai, Y. and Judd, K. L. (2013). Advances in Numerical Dynamic Programming and New Applications. In Schmedders, K. and Judd, K. L., editors, *Handbook of Computational Economics*. Elsevier.

Dubé, J.-P. H., Fox, J. T., and Su, C.-L. (2012). Improving the Numerical Performance of Static and Dynamic Aggregate Discrete Choice Random Coefficients Demand Estimation. *Econometrica: Journal of the Econometric Society*, 80(5):2231–2267.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization*. Academic Press.

Grüne, L. (1997). An Adaptive Grid Scheme for the Discrete Hamilton-Jacobi-Bellman Equation. *Numerische Mathematik*, 75(3):319–337.

Grüne, L. and Semmler, W. (2004). Using Dynamic Programming with Adaptive Grid Scheme for Optimal Control Problems in Economics. *Journal of Economic Dynamics and Control*, 28(12):2427–2456.

Hendel, I. and Nevo, A. (2006). Measuring the Implications of Sales and Consumer Inventory Behavior. *Econometrica: Journal of the Econometric Society*, 74(6):1637–1673.

Jäckel, P. (2005). A Note on Multivariate Gauss-Hermite Quadrature. Technical report.

Judd, K. L. (1998). *Numerical Methods in Economics*. The MIT Press.

Keane, M. P., Todd, P. E., and Wolpin, K. I. (2011). The Structural Estimation of Behavioral Models: Discrete Choice Dynamic Programming Methods and Applications. In Ashenfelter, O. and Card, D., editors, *Handbook of Labor Economics*, pages 331–461. Elsevier.

Keane, M. P. and Wolpin, K. I. (1994). The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence. *The Review of Economics and Statistics*, 76(4):648–672.

Kythe, P. K. and Schäferkotter, M. R. (2005). *Handbook of Computational Methods for Integration*, volume 1. CRC Press.

Larsen, B. J., Oswald, F., Reich, G., and Wunderli, D. (2012). A Test of the Extreme Value Type I Assumption in the Bus Engine Replacement Model. *Economics Letters*, 116(2):213–216.

McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. In Zarembka, P., editor, *Frontiers in Econometrics*, pages 105–142. Academic Press.

McFadden, D. (1981). Econometric Models for Probabilistic Choice. In Manski, C. F. and McFadden, D., editors, *Structural Analysis of Discrete Data with Econometric Applications*, pages 198–272. The MIT Press.

Melnikov, O. (2012). Demand for Differentiated Durable Products: The Case of the U.S. Computer Printer Market. *Economic Inquiry*, 51(2):1277–1298.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer Science+Business Media.

Norets, A. (2009). Inference in Dynamic Discrete Choice Models with Serially Correlated Unobserved State Variables. *Econometrica: Journal of the Econometric Society*, 77(5):1665–1682.

Norets, A. (2012). Estimation of Dynamic Discrete Choice Models Using Artificial Neural Network Approximations. *Econometric Reviews*, 31(1):84–106.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition.

Rust, J. (1987). Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica: Journal of the Econometric Society*, 55(5):999–1033.

Rust, J. (1988). Maximum Likelihood Estimation of Discrete Control Processes. *SIAM Journal on Control and Optimization*, 26(5):1006–1024.

Rust, J. (1996). Numerical Dynamic Programming in Economics. In Amman, H. M., Kendrick, D. A., and Rust, J., editors, *Handbook of Computational Economics*, pages 619–729. Elsevier.

Schenk, O. and Gärtner, K. (2004). Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO. *Future Generation Computer Systems*, 20(3):475–487.

Stinebrickner, T. R. (2000). Serially Correlated Variables in Dynamic, Discrete Choice Models. *Journal of Applied Econometrics*, 15(6):595–624.

Su, C.-L. and Judd, K. L. (2012). Constrained Optimization Approaches to Estimation of Structural Models. *Econometrica: Journal of the Econometric Society*, 80(5):2213–2230.

Sullivan, P. (2006). Interpolating Value Functions in Discrete Choice Dynamic Programming Models. Technical report.

Wächter, A. and Biegler, L. T. (2005). On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57.